A picture containing drawing

Description automatically generated

**PLAN FEZABILITATE**

ȘI EXECUȚIE A APLICAȚIEI DE MONITORIZARE A VEHICULELOR MEDIA - ***ADCODE***

**1. Desriere generală**

**1.1 Structura**, conform specificațiilor cerute de beneficiar această aplicație va avea o arhitectură de tip client-server formata din modulul expus pe dispozitivul clientului, denumit in continuare **frontend** și modulul resident pe server, modul a carui instalare și operare nu este inclusă in prezentul document, denumit în continuare **backend**. De asemenea din modulul de **frontend** va fi derivateă cu ajutorul aplicției **Cordova** o variant mobile a aplicției, atât pentru dispositive de tip **Android** cât și **iOS**.

**1.2 Tipurile de conturi utilizatori** sunt enumerate și descries mai jos:

1. **Cont TALENT** – orice entitate, persoană juridică sau fizică, care, prin activitatea sa, generează drepturi intelectuale care sunt inglobate intr-un vehicol media, monitorizat de aplicația de față.(ex: actori, actori de voce, figuratie special, regizori, scenariști, etc)
2. **Cont PRODUCĂTOR –** orice entitate, persoană juridică sau fizică, care, prin activitatea sa generează (produce) vehicole media. (es: case de producție, studiouri, agentii ce au resurse in-house pentru producție audio/video, companii care dețin astfel de facilități, inclusive posture de radio și televiziune ce realizează producții propria, etc)
3. **Cont EMIȚĂTOR –** orice persoană juridică ce prin natura activității difuzează, face public, expune producțiile audio/video menționate anterior și obține în mod direct sau collateral profit din axeastă activitate. (posturi de radio, televiziuni, resturante, terase, etc)
4. **Cont ARBITRU -** orice persoană juridică ce prin natura activității sale este responsabilă de colectarea și redistribuirea de drepturi bănești aferente drepturilor intelectuale generate de **TALENTE** (cum sunt organismele de gestiune colectiva) sau alte entități în a caror activitate este necesară

**1.3. Specificații generale – GRAPHQL**, a fost soluția adoptată în ceea ce privește gestiunea transferului de data spre și dinspre server la modulul client. Avantajul de a transfera numai datele de care aplicația are nevoie la un moment dat a prevalat indiscutabil în detrimental altor soluții. Este estimate un volum foarte mare de dtate ce vor fi gestionate la nivelul server-ului și a transfera maim ult decât este strict necesar pentru funcționarea punctualăa aplicației, ar putea adduce neajunsuri majore parametrilor de funcționare a aacesteia.

De altfel, această tehnologie, dezvoltată tot de departamentele de proiectare de software ale celor de la Facebook/Instagram/Whatsapp este deja utilizată cu success în aceste aplicații de success, cee ace denotă o robustețe notabilă, în condițiile utilizării la scară larga în prezența unui volum masiv de date.

**2. Descrierea componentelor aplicației**

**2.2 Frontend**-ul este acea parte a aplicatiei web cu care utilizatorul interacționează și este constituită suma elementelor vizuale ce facilitează această interacțiune plus codul necesar operării acestora. Realizarea acestui modul presupune două faze generale:

1. design-ul/prototyping-ul (dezvoltarea componentelor vizuale ale interfeței, așezarea acestora și fluxul utilizator)
2. implemetarea logicii interfetei (implementare HTML, CSS si JavaScript).

**2.2.1 Designul**, conform specificațiilor beneficiarului, designul aplicației va fi unul axat pe o experiență cât mai fluidă a utilizatorului. Elementele grafice se vor axa pe simplitate, designul va fi unul minimalist, actual, bazat pe doua culori principale, plus variații ale acestora, utilizate în evidențierea aspectelor funcționale.

Implementarea CSS se va face cu ajutorul librăriei Tailwind, o librarie axată pe funcționalitate, cu grad mare de reutilizare funcțională.

**2.2.2 Funcționalitatea** se va baza pe libraria **React**, o librărie open source gestionată și constat îmbunătățită de catre una din cele mai prestigioase echipe de programatori din lume și anume cea de la **Facebook/Instagram/Whatsapp.** Această librărie a fost aleasă ca bază de pornire pentru funcționalitatea interfeței client pentru stabilitatea dovedită , continua îmbunatațire și întreținere a bazei de cod și, mai ales, datorită continuității în evoluție, cu evitarea unor situaiți fracturare a compatibilității istorice. Tocmai pentru că aplicația de față se află într-o fază de început, urmând să se dezvolte rapid în etapele următoare, am ales această librărie, ce va permite incrementarea rapida a facilităților oferite, pe o bază de cod stabilă, întreținută și constant adusă la zi.

**2.2.3 Librării publice open-source** utilizate parte dinecosistemul **JavaScript** ce se bazează pe **npm** (Node Package Manager), un mediu de gestiune de pachetele și librării în mediul de executie Node.js si care se bazeaza pe un client de linie de comanda si o baza de date formata din pachete publice si private cunoscută sub numele de **npm registry.**

Dezvoltarea aplcației de față va utiliza o selecție de librării accesibile în acest ecosystem. Criteriile de selecție au fost:

* compatibilitate
* constanța mentenanței
* numarul de colaboratori activi
* numărul de utilizări
* referințele utilizatorilor pe site-uri de specialitate (medium, stack-overflow)

De asemeni, desi specific ecosistemului este managerul de packete numit **npm**, am optat pentru alternativa dezvoltată de aceeași echipă ca și librăria **React**, menționată mai sus și anume aplicația de management al pachetelor **yarn**, care rezolvă unele dintre neajunsurile **npm** păstrând funcționalitățile si aceeasi structura acestuia.

Din librariile open-source puse la dispoziție de **npm-registry** am selectaturmătoarele:

1. react – versiunea 16.13.1

*librăria specificată mai sus ce stă la baza funcționalității aplicației*

1. base64-img – versiunea 1.0.4

*o librărie utilitatră pentru encodare a imaginilor în format base64*

1. react-dom – versiunea 16.13.1

*utilitarul asociat librăriei React pentru dezvoltarea de aplicații în browser*

1. classnames – versiunea 2.2.6

*o librărie utilitatră pentru aplicarea de clase css condiționale*

1. formik – versiunea 2.1.5

*o librărie utilitatră pentru gestiune formularelor specifică librăriei* ***React***

1. react-autosuggest – versiunea 10.0.2

*o librărie utilitatră pentru meniuri editabile ce ofera sugestii de selecție*

1. react-avatar-editor – versiunea 11.0.9

*o librărie utilitatră pentru deitarea de imagini de tip* ***avatar*** *asociate profilului utilizator*

1. react-debounce-input - – versiunea 3.2.2

*o librărie utilitatră pentru temporizarea validării asyncrone în formulare*

1. react-dropzone – versiunea 11.1.0  
   *o librărie utilitatră pentru incarcarea fișierelor cu tehnica „drag&drop”*
2. react-router – versiunea 5.2.0

*o librărie utilitatră pentru realizrii navigației în cadrul applicației*

1. react-router-dom – versiunea 5.2.0  
   *utilitarul asociat librăriei* ***react-router*** *pentru dezvoltarea de aplicații în browser*
2. react-scripts – versiunea 3.4.1

*o librărie utilitatră pentru encapsularea aplicțiilor* ***React*** *în formatul* ***CRA***

1. url-join – versiunea 4.0.1

*o librărie utilitatră pentru concatenarea componentelor de tip șir de caractere ce compun o adresa url*

1. yup – versiunea 0.29.1

*o librărie utilitatră pentru validarea de formulare*

**2.2.4 Librăriile specifice dezvoltate de către furnizor** sunt enumerate mai jos:

**1**.**Splash Library** - *o librărie utilitatră cu înglobează două funcționalități:*

1. *aceea de a prezenta o grafică prezentațională în timp ce în planul secundar se încarcă aplicația.*
2. *Supune utilizatorului spre aprobare politica aplicației în cee ace privește politica GDPR-ul si politica de acceptare a cookie-urilor atunci cand un utilizator acceseaza aplicatia web pentru prima data.*

Opțiunila utilizatorului de a permite rulareea aplicației pe dispozitivul său este salvata la nivelul **localStorage** în browserul acestuia respective în registrul specific în cazul aplicației mobile.

**2. Strings Library** - *o librărie folosită pentru suportul multilingual al aplicatiei.*

Această librărie permite modificarea limbii în care este afiștă aplicația în orice moment execuție

**3. Status Library** – *o librărie ce gestionează mesajele legate de starea aplicației*

Dupa cum sugereaza si numele, aceasta librarie va citi statusul erorilor sau defectiunilor care vor aparea in timp ce utilizatorul va folosi aceasta aplicatie si vor aparea mesaje sugestive pentru a face utilizatorul sa inteleaga ce se intampla și cum poate rezolva rapid problemele intampinate

Iata, mai jos un esantion din ecranul de start:Graphical user interface, application

Description automatically generated

Această imagine a fost adăugată ca exemplu. Următoarele versiuni ale aplicției vor modifica și îmbunătăți această resursă

**4. Auth Library** - *o librărie ce gestionează aspectele legate de autentificarea utilizatorilor*

Aceasta librarie este responsabilă de inregistrarea utilizatorilor, de creerea de conturi și de ieșrea acestora din aplicație. De asemea această librărie pune la dispoziția aplicației datele utilizatoruilui current și gestioneaza sesiunea curentă.

**5. Http Library** - *o library care face legatura cu ajutorul protocolului* ***http*** *dintre* ***frontend*** *si* ***backend*.**

Această librărie uniică in cadrul unui singur serviciu toate nevoile aplicației de comunicare ale aplicației între cele două componente, punănd la dispoziția tutror consumatorilor acestui serviciu trei tipuri de acces:

1. http normal – accesul la punctele de acces ale backendului in mod standard, utilizat, de exemplu, de serviciu de autentificare la înregistrare în aplicație
2. http graphql – accesul la punctul de acces de date de tip graphql. Acest mod de acces este cel mai present în cadrul aplicației și este principal canal de hidratare a datelor pe modulul client (**frontend**)
3. http origin – accesul la radacina de servire a modulului client. Acesta este utilizat, de exemplul, in cazul în care este schimbata limba de afisare a aplicatiei, fișierele ce contin textile specifice fiecarei limbi, aflându-se intr-un subdirector al originii modulului client

Deși există librării similar în ecosistemul open source, am optat pentru varianta dezvoltării acestor librării de la 0, special pentru aplicația de față din următoarele considerente:

1. **Durata de mentenanță**. - Ecosistemul open-source este un mediu generat în mare parte, de voluntari, In timpul lor liber. În funcție de oportunitățile ce apar în viața lor, de multe ori, renunță să mai continue mentenanța la pachetele publicate, în principal, din lipsă de timp. De aceea, cu excepția unor pachete întreținute active de entități mari (Facebook, Google, Amazon), pachetele mici prezintă un grad de risc, în sensul în care, să spunem, React trece l o versiune superioară ce necesită un update ale unor pachete mai mici, ce depind de acesta și pe care noi le utilizam în aplicaia noastră. Din lipsă de support active, pachetul nu mai este updatat și în aplicația noastră apare o eroare ce nu ține de codul propriu , ci datorită acestui pachet.
2. **Complexitatea indusă –** Foarte multe pachete sunt completate cu optiuni și utilități ale diferiților administrator sau colaboratori ce vin să adauge astfel de funcționalități unui pachet, ajungând astfel ca, în scurt timp, acesta să crescă în complexitate și volum, mult sub optimul, pe care îl cere aplicația noastră. Aceste opțiuni pe care, fiecare collaborator il adaugă, pot foarte ușor să inducă erori și să crească previzibilitatea
3. **Dificultate de dezvoltare –** In viitor aplicția de față va necesita o dezvoltare în toate categoriile de funcționalități pe care le oferă, de aceea controlul la nivelul librăriilor de bază este absolut esențial. Aceste librării fiind executate specific pentru a deservi o astfel de aplicație, sunt ușor scalabile, iar efectele la nivelul global sunt previzibile și ușor de depanat.